ECE331 Handout 7: Microcontroller Peripheral Hardware

MCS12DP256B Block Diagram: Peripheral I/O Devices associated with Ports

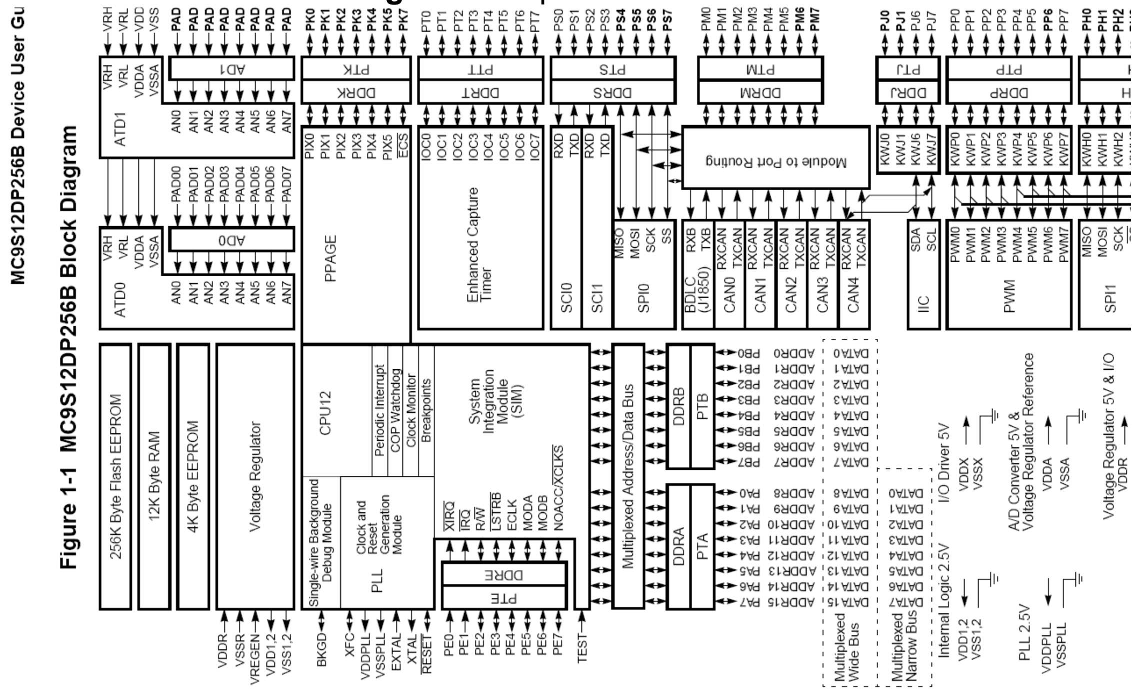
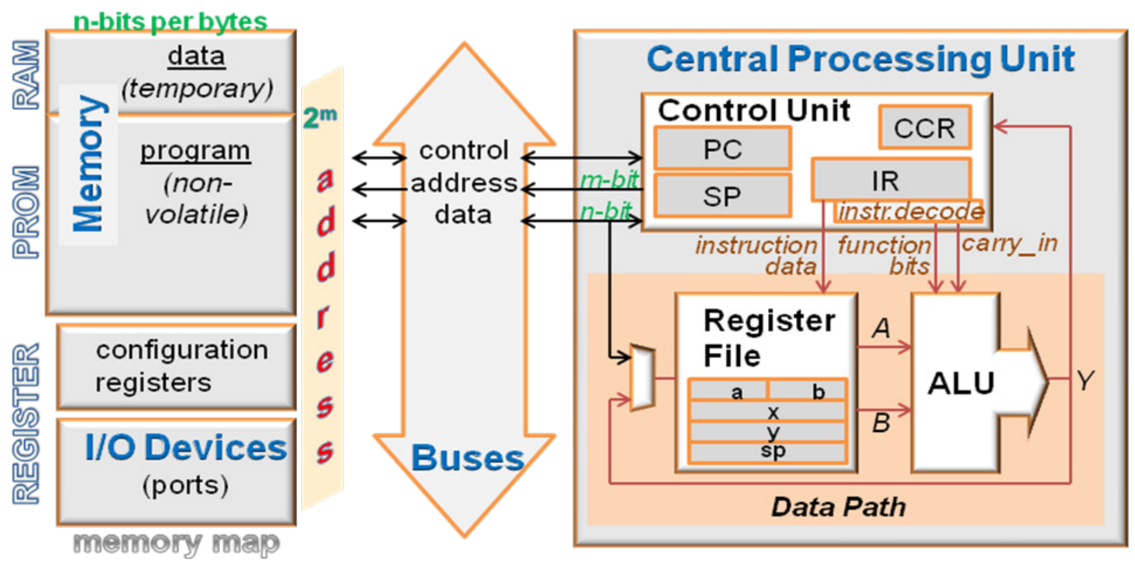

MC9S12DP256B Device User Gu

Figure 1-1 MC9S12DP256B Block Diagram

Expanded Microcontroller Architecture: CPU and Peripheral Hardware Details

- CPU
  - components and control signals from Control Unit
  - connections between Control Unit and Data Path
  - components and signal flow within data path
- Peripheral Hardware Memory Map
  - Memory and I/O Devices connected through Bus
  - all peripheral blocks mapped to addresses so CPU can read/write them
  - physical memory type varies (register, PROM, RAM)



Memory Map: 68HCS12 CPU and MC9S12DP256B Evaluation Board

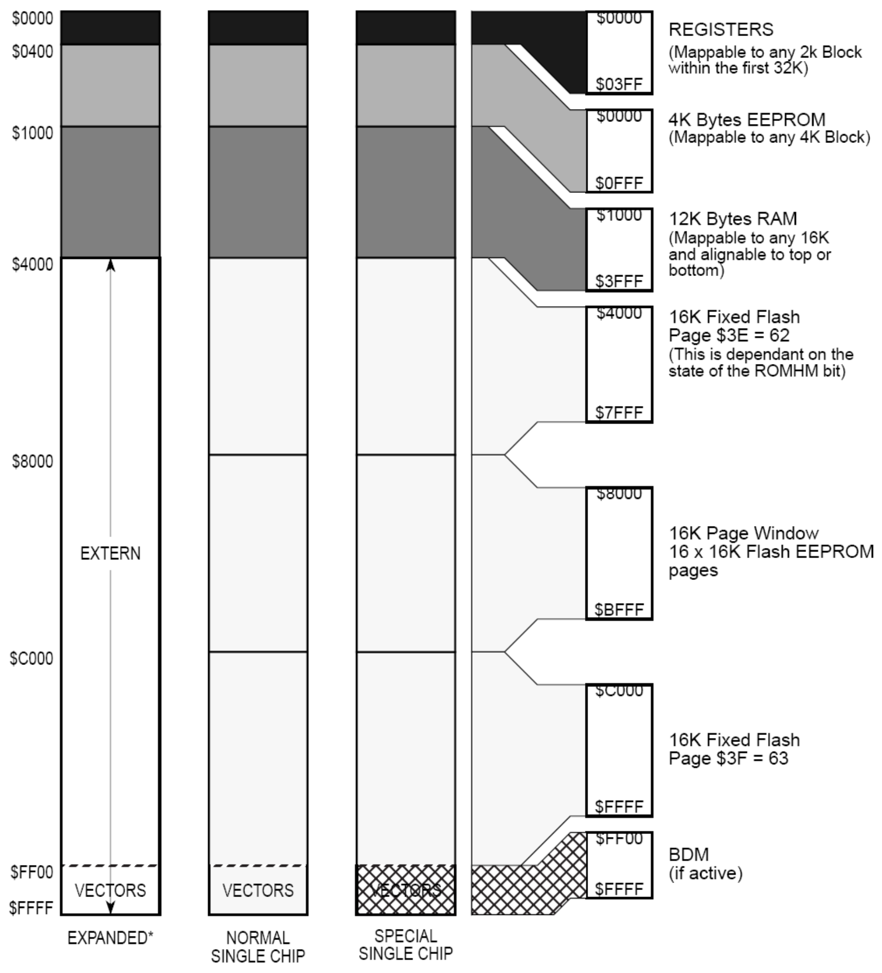
Configuration Register Set

Address	Module	Size (Bytes)
\$0000 - \$0017	CORE (Ports A, B, E, Modes, Inits, Test)	24
\$0018 - \$0019	Reserved	2
\$001A - \$001B	Device ID register (PARTID)	2
\$001C - \$001F	CORE (MEMSIZ, IRQ, HPRIO)	4
\$0020 - \$0027	Reserved	8
\$0028 - \$002F	CORE (Background Debug Mode)	8
\$0030 - \$0033	CORE (PPAGE, Port K)	4
\$0034 - \$003F	Clock and Reset Generator (PLL, RTI, COP)	12
\$0040 - \$007F	Enhanced Capture Timer 16-bit 8 channels	64
\$0080 - \$009F	Analog to Digital Converter 10-bit 8 channels (ATD0)	32
\$00A0 - \$00C7	Pulse Width Modulator 8-bit 8 channels (PWM)	40
\$00C8 - \$00CF	Serial Communications Interface 0 (SCI0)	8
\$00D0 - \$00D7	Serial Communications Interface 0 (SCI1)	8
\$00D8 - \$00DF	Serial Peripheral Interface (SPI0)	8
\$00E0 - \$00E7	Inter IC Bus	8
\$00E8 - \$00EF	Byte Data Link Controller (BDLC)	8
\$00F0 - \$00F7	Serial Peripheral Interface (SPI1)	8
\$00F8 - \$00FF	Serial Peripheral Interface (SPI2)	8
\$0100- \$010F	Flash Control Register	16
\$0110 - \$011B	EEPROM Control Register	12
\$011C - \$011F	Reserved	4
\$0120 - \$013F	Analog to Digital Converter 10-bit 8 channels (ATD1)	32
\$0140 - \$017F	Motorola Scalable Can (CAN0)	64
\$0180 - \$01BF	Motorola Scalable Can (CAN1)	64
\$01C0 - \$01FF	Motorola Scalable Can (CAN2)	64
\$0200 - \$023F	Motorola Scalable Can (CAN3)	64
\$0240 - \$027F	Port Integration Module (PIM)	64
\$0280 - \$02BF	Motorola Scalable Can (CAN4)	64
\$02C0 - \$03FF	Reserved	320

Physical Memory

\$0000 - \$0FFF	EEPROM array	4096
\$1000 - \$3FFF	RAM array	12288
\$4000 - \$7FFF	Fixed Flash EEPROM array incl. 0.5K, 1K, 2K or 4K Protected Sector at start	16384
\$8000 - \$BFFF	Flash EEPROM Page Window	16384
\$C000 - \$FFFF	Fixed Flash EEPROM array incl. 0.5K, 1K, 2K or 4K Protected Sector at end and 256 bytes of Vector Space at \$FF80 - \$FFFF	16384

Figure 1-2 MC9S12DP256B Memory Map



MON12 and NOICE Memory Map

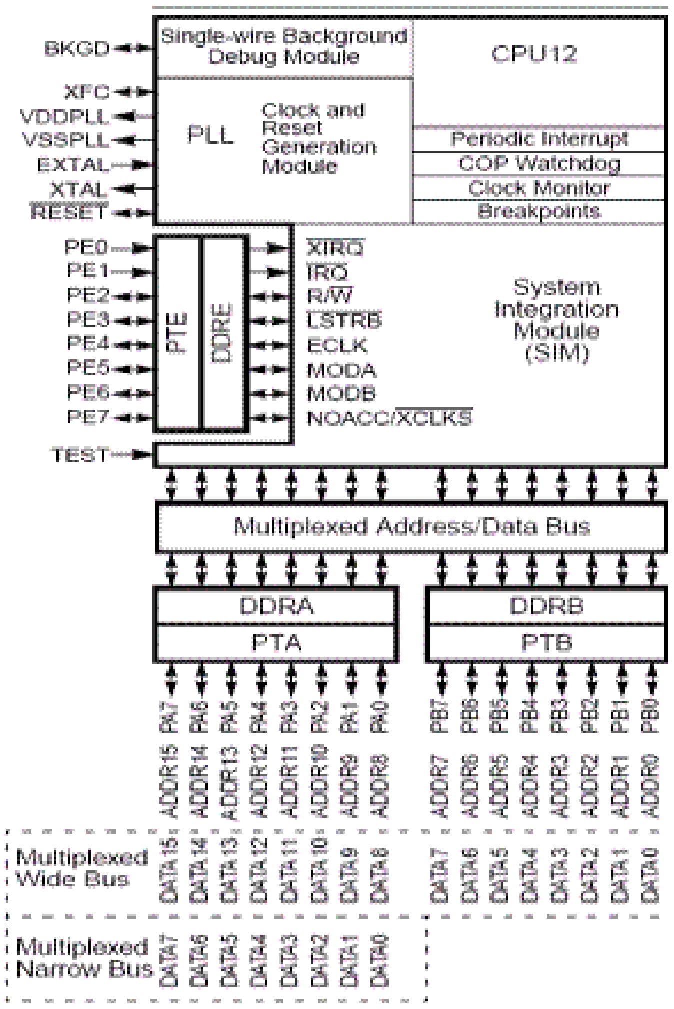
ADDRESS	TYPE MEMORY	MEMORY APPLICATION
$C000 - $FFFF	FLASH	MON12, NOICE, and Utility firmware located in internal flash, Page $3F.
$8000 - $BFFF	External Ram	User Paged Program Memory space, pages $20 - $2E. Note: Pages $30 - $3F reside in the internal flash.
$4000 - $7FFF	External Ram	User Program Memory, emulate fixed page $3E.
$3F8C - $3FFD	Internal Ram	Ram Interrupt Vector Table
$3E00 - $3F8B	Internal Ram	Monitor reserved ram memory. Stacks and variables.
$1000 - $3DFF	Internal Ram	User Internal Ram memory
$0400 - $0FEB	Internal EEprom	User EEprom memory, Monitor reserves $FEC - $FEF for Autostart, user should avoid $FF0 - $FFF memory use.
$0000 - $03FF	HCS12 Registers	Monitor or user access to control registers.

HCS12 Modes of Operation

MODC	*MODB*	*MODA*	*Mode*	*Port A*	*Port B*
0	0	0	special single chip	G.P. I/O	G.P. I/O
0	0	1	special expanded narrow	Addr/Data	Addr
0	1	0	special peripheral	Addr/Data	Addr/Data
0	1	1	special expanded wide	Addr/Data	Addr/Data
1	0	0	normal single chip	G.P. I/O	G.P. I/O
1	0	1	normal expanded narrow	Addr/Data	Addr
1	1	0	reserved	--	--
1	1	1	normal expanded wide	Addr/Data	Addr/Data

G.P. = general purpose

HCS12 Ports for Expanded Modes



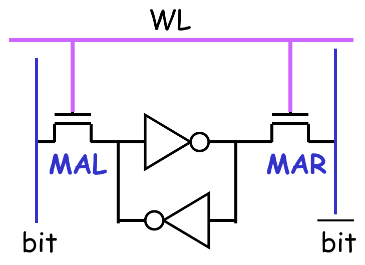
BKGD
Single-wire Background Debug Module
CPU12
XFC
VDDPLL
VSSPLL
EXTAL
XTAL
RESET
PLL
Clock and Reset Generation Module
Periodic Interrupt
COP Watchdog
Clock Monitor
Breakpoints
PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
PTE
DDRE
XIRQ
IRQ
R/W
LSTRB
ECLK
MODA
MODB
NOACC/XCLKS
System Integration Module (SIM)
TEST
Multiplexed Address/Data Bus
DDRA
PTA
DDRB
PTB
PA7 PA6 PA5 PA4 PA3 PA2 PA1 PA0
PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0
ADDR15 ADDR14 ADDR13 ADDR12 ADDR11 ADDR10 ADDR9 ADDR8
ADDR7 ADDR6 ADDR5 ADDR4 ADDR3 ADDR2 ADDR1 ADDR0
Multiplexed Wide Bus
DATA15 DATA14 DATA13 DATA12 DATA11 DATA10 DATA9 DATA8
DATA7 DATA6 DATA5 DATA4 DATA3 DATA2 DATA1 DATA0
Multiplexed Narrow Bus
DATA7 DATA6 DATA5 DATA4 DATA3 DATA2 DATA1 DATA0

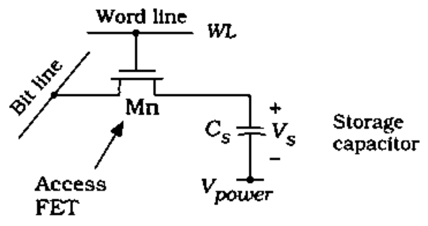
Memory Basics

- RAM: Random Access Memory
  - historically defined as memory array with individual bit access
  - refers to memory with both Read and Write capabilities
- ROM: Read Only Memory
  - no capabilities for "online" memory Write operations
  - Write typically requires high voltages or erasing by UV light
- Volatility of Memory
  - volatile memory loses data over time or when power is removed
    - RAM is volatile
  - non-volatile memory stores date even when power is removed
    - ROM is non-volatile
- Static vs. Dynamic Memory
  - Static: holds data as long as power is applied (SRAM)
  - Dynamic: will lose data unless refreshed periodically (DRAM)

SRAM/DRAM Basics

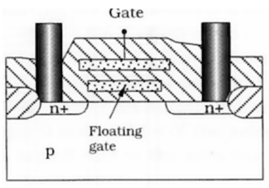
- SRAM: Static Random Access Memory
  - Static: holds data as long as power is applied
  - Volatile: can not hold data if power is removed
  - 3 Operation States: hold, write, read
  - Basic 6T (6 transistor) SRAM Cell
    - bistable (cross-coupled) INVs for storage
    - access transistors MAL & MAR
    - word line, WL, controls access
      - WL = 0 (hold) = 1 (read/write)
- DRAM: Dynamic Random Access Memory
  - Dynamic: must be refreshed periodically
  - Volatile: loses data when power is removed
  - 1T DRAM Cell
    - single access transistor; storage capacitor
    - control input: word line (WL); data I/O: bit line
- DRAM to SRAM Comparison
  - DRAM is smaller & less expensive per bit
  - SRAM is faster
  - DRAM requires more peripheral circuitry





ROM/PROM Basics

- ROM: Read Only Memory
  - no capabilities for "online" memory Write operations
  - data programmed
    - during fabrication: ROM
    - with high voltages: PROM
    - by control logic: PLA
  - Non-volatile: data stored even when power is removed
- PROM: Programmable Read Only Memory
    - programmable by user -using special program tools/modes
    - read only memory -during normal use
    - non-volatile
  - Read Operation
    - like any ROM: address bits select output bit combinations
  - Write Operation
    - typically requires high voltage (~15V) control inputs to set data
      - stores charge to floating gate (see figure) to set to Hi or Low
  - Erase Operation
    - to change data
    - EPROM: erasable PROM: uses UV light to reset all bits
    - EEPROM: electrically-erasable PROM, erase with control voltage



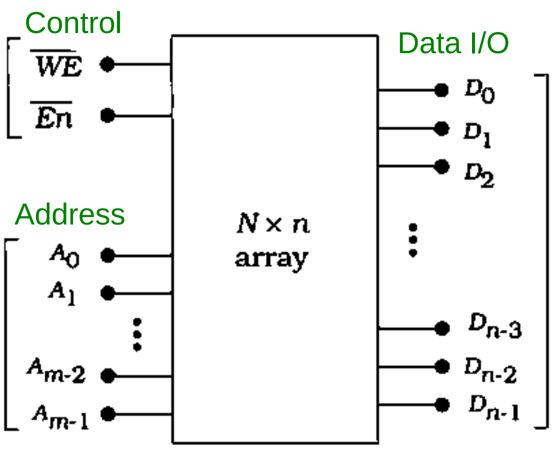
EPROM device structure

Comparison of Memory Types

- DRAM
  - very high density → cheap data cache in computers
  - must be periodically refreshed → slower than SRAM
  - volatile; no good for program (long term) storage
- SRAM (basically a Latch)
  - fastest type of memory
  - low density → more expensive
    - generally used in small amounts (L2 cache) or expensive servers
- EEPROM
  - slow/complex to write → not good for fast cache
  - non-volatile; best choice for program memory
- ROM
  - hardware coded data; rarely used except for bootup code
- Register (flip flop)
  - functionally similar to SRAM but less dense (and thus expensive)
  - reserved for data manipulation applications

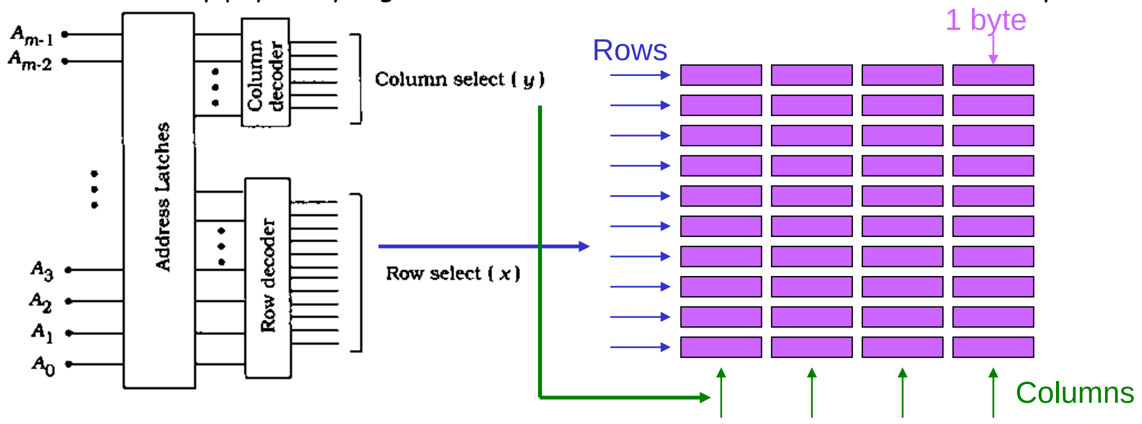
Memory Arrays

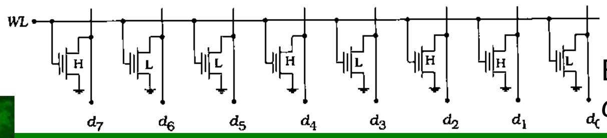
- N x n array of 1-bit cells
  - n = byte width; 8, 16, 32, etc.
  - N = number of bytes
  - m = number of address bits
    - max N = 2^m
- Array I/O
  - data (in and out)
    - D_{n-1} - D_0
  - address
    - A_{m-1} - A_0
  - control
    - varies with design
    - WE = write enable (assert low)
      - WE=1=read, WE=0=write
    - En = block enable (assert low)
      - used as chip enable (CE) for an SRAM chip



Memory Array Addressing

- Standard Memory Addressing Scheme
  - ***m*** address bits are divided into ***x*** row bits and ***y*** column bits (x+y=m)
    - address bits are encoded so that 2^m = N
    - array physically organized with both vertical and horizontal stacks of bytes

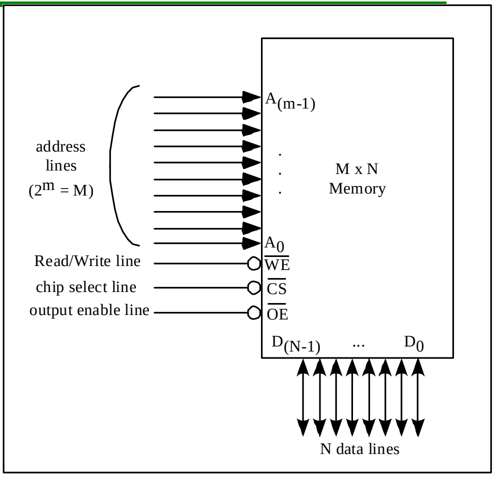


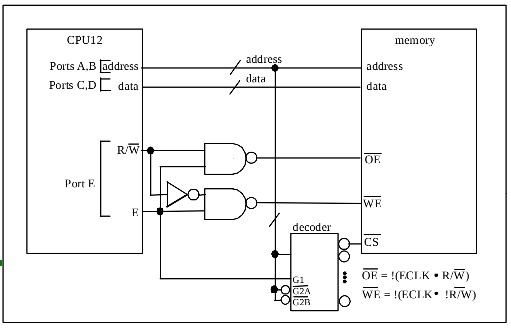


Example byte:
one word in an 8b-wide EPROM

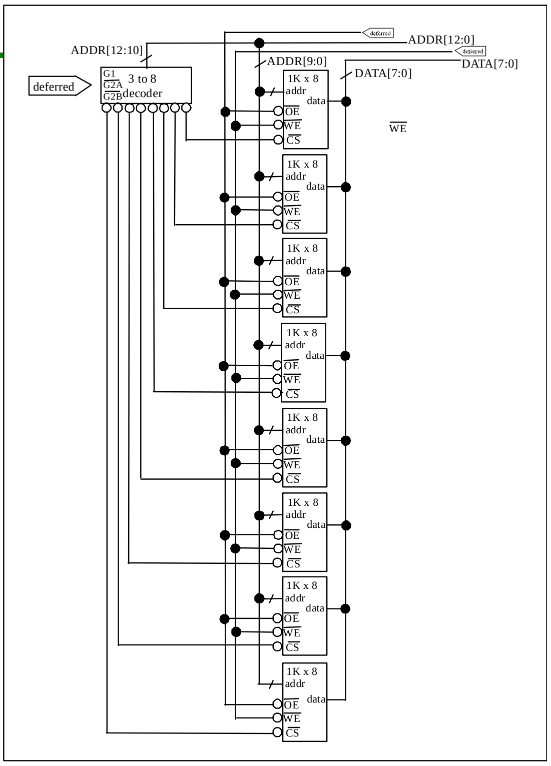
Typical Memory Chip

- Data
  - x-bits in parallel, typically x = 8, 16
- Address signals
  - m address signals → $M=2^m$ addresses
- Control signals
  - **/WE: write enable** - when activated, values on data lines are written to specified address
  - **/OE: output enable** - data at specified location placed on data pins of memory chip, data lines connected to data bus using tristate outputs
  - **/CS: chip select** - selects a specific chip in an array of memory chips
- Connection to HC12 -----→

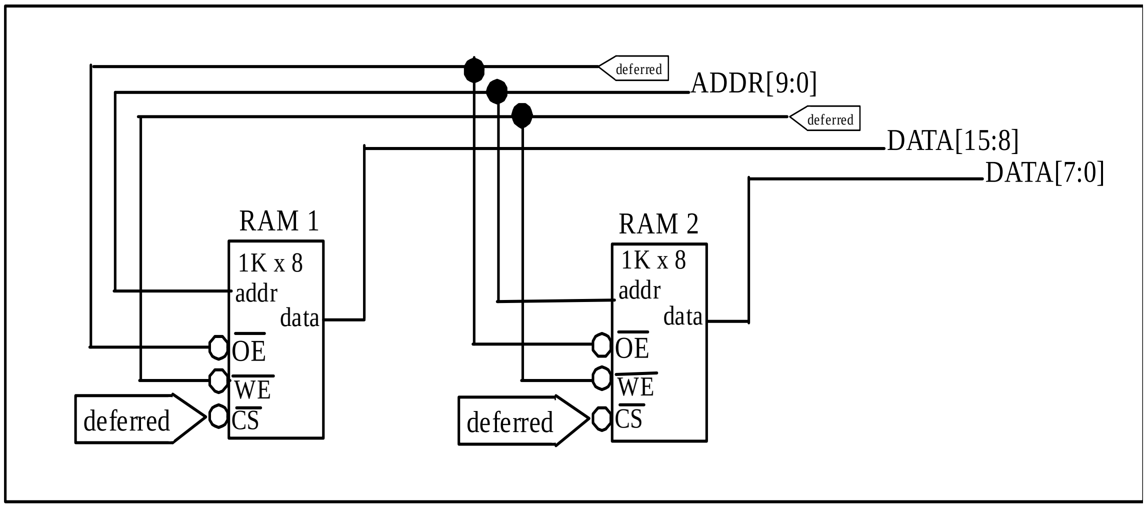




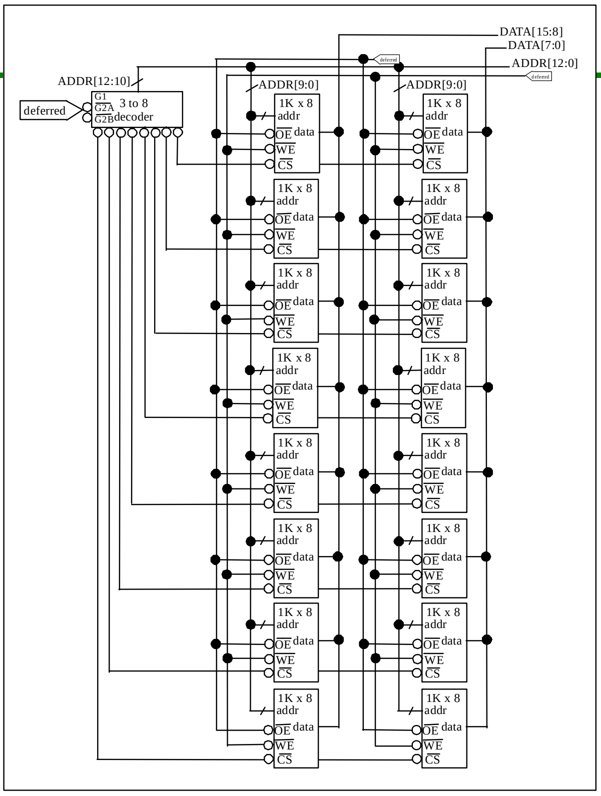
Memory Expansion
expanding memory length



Memory Expansion
expanding memory width



Memory Expansion
expanding memory length and width


DATA[15:8]
DATA[7:0]
ADDR[12:0]
deferred
deferred
ADDR[12:10]
ADDR[9:0]
ADDR[9:0]
deferred
G1
G2A
G2B
3 to 8
decoder
1K x 8
addr
data
OE
WE
CS